2025-07-18 Tensor Network 2025

テンソルネットワーク形式でのモンテカルロ法
Monte Carlo method in TN representation

Synge Todo / 藤堂眞治 <wistaria@phys.s.u-tokyo.ac.jp>
Department of Physics, University of Tokyo

arXiv:2412.02974

Recent activities of our group

- **Development of quantum simulation and quantum embedding methods using tensor networks and sampling**
  - **Quantum simulation / 量子シミュレーション**
    - State preparation for quantum simulation → arXiv:2506.04663
    - Mitigation of the negative sign problem in quantum Monte Carlo → arXiv:2501.18069
  - **Tensor network algorithms / テンソルネットワークアルゴリズム**
    - Quantum states of many-body systems → arXiv:2403.11490
    - Applications to field theory → arXiv:2410.09485, arXiv:2501.18918
    - Compression of generative models → arXiv:2408.10669, arXiv:2504.06722
    - Tensor networks for option pricing → arXiv:2405.00701, arXiv:2507.08482
    - Tensor networks + MCMC simulator → arXiv:2412.02974
  - **Quantum embedding / 量子埋め込み**
    - Sample complexity of matrix product states at finite T → arXiv:2403.10018
    - Embedding tensor networks in quantum circuits → arXiv:2501.18856, arXiv:2504.09250, arXiv:2504.14995
    - Optimization of tensor contraction order by graph theory
  - **QEC, error mitigation, quantum compilation / 量子エラー訂正、エラー緩和、量子コンパイル**
    - Tensor network decoder, noise model estimation → arXiv:2406.08981
    - Decomposition of multi-controlled gates → arXiv:2109.13223, arXiv:2410.00910
    - Automatic differentiation of parameterized quantum circuits
    - Load/Store architecture for limited-scale FTQC → arXiv:2412.20486

arXiv:2307.02284, to appear in PRR tonight

Statistical error in MCMC measurements

arXiv:2412.02974

- There is autocorrelation between successive configurations

$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\mathrm{int}}}{M}$$

  - σ_0^2 : population variance (variance of time-series data)
  - M : number of Monte Carlo steps
  - τ_{int} : autocorrelation time (determined by the MC dynamics)
  - effective sample size → $M/2\tau_{\mathrm{int}}$

- For systems with a negative sign problem

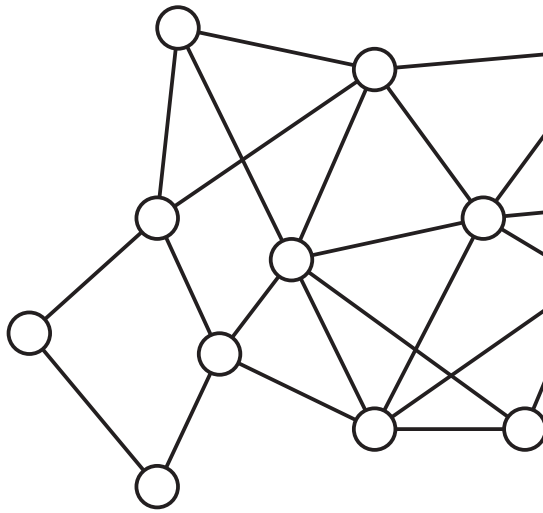
$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\mathrm{int}}}{M\, s^2}$$

  - frustrated quantum spin systems, fermionic systems, unitary evolution, etc
  - s : average sign (exponentially small for larger system, lower temperature, longer time)
  - effective sample size → $s^2 M/2\tau_{\mathrm{int}}$

Advances in Markov chain Monte Carlo

arXiv:2412.02974

- Representation (definition of “configurations” and “weighs”)
  - path integral representation for quantum Monte Carlo (1976), Bayesian inference (1990)...
- Choice of ensemble
  - extended ensemble method: multicanonical MC (1991, 2001), exchange MC (1996), lifting (2000)...
- Generation of set of candidate configurations
  - non-local (cluster) updates: Swendsen-Wang (1987), Hamiltonian MC (1987), loop (1993), worm (1998)...
- Choice of transition kernel (probabilities)
  - Metropolis, heat bath (Gibbs sampler), over-relaxation (1987), irreversible kernel (2010), event-chain (2013)...
- Algorithm for generating a configuration according to transition probabilities
  - N-fold way (rejection free) (1975), Walker’s method (1977, 2019), order-N algorithm (1995, 2009)...



Reduction of population variance

arXiv:2412.02974

- Can we change (or control) population variance? $\sigma^2 = \dfrac{2\sigma_0^2 \tau_{\mathrm{int}}}{M}$
  - Standard textbooks say...

$$C = \frac{\beta^2}{N}\left[\langle E^2\rangle - \langle E\rangle^2\right] \quad \Rightarrow \quad \mathrm{Var}[E] = \frac{NC}{\beta^2}$$

  - The variance is given by the specific heat (= physical property of the system)
  - → not affected by the details of the sampling scheme?

- For $Z = \sum_s W(\beta, s)$, energy and specific heat are given by

$$E = -\frac{\partial Z/\partial\beta}{Z} = -\left\langle \frac{1}{W}\frac{\partial W}{\partial\beta}\right\rangle \qquad C = -\frac{\beta}{N}\frac{\partial E}{\partial\beta} = \frac{\beta^2}{N}\left(\left\langle \frac{1}{W}\frac{\partial^2 W}{\partial\beta^2}\right\rangle - \left\langle \frac{1}{W}\frac{\partial W}{\partial\beta}\right\rangle^2\right)$$

  - In general, $\dfrac{1}{W}\dfrac{\partial^2 W}{\partial\beta^2} \neq \left(\dfrac{1}{W}\dfrac{\partial W}{\partial\beta}\right)^2$
  - above statement is valid only when the weight is given by

$$W(\beta, s) = \exp[-\beta E(s)]$$

Reduction of population variance

arXiv:2412.02974

$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\mathrm{int}}}{M}$$

- Generally, population variance is determined when we choose a representation of the target partition function
  - e.g.

$$Z = \sum_s \exp[-\beta E(s)]$$

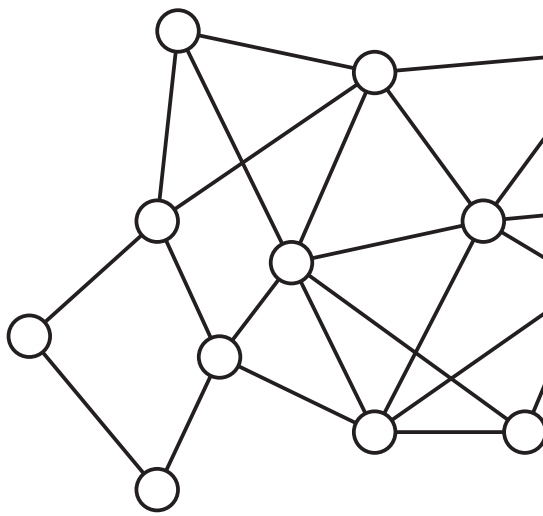
- Some attempts to reduce σ_0^2
  - Improved estimators in cluster algorithm
    - partition function and physical quantities (magnetizationˆ2, etc) are defined in term of cluster configurations

- No systematic approach has been proposed so far!

Advances in Markov chain Monte Carlo

arXiv:2412.02974

- Representation (definition of “configurations” and “weighs”)
  - path integral representation for quantum Monte Carlo (1976), Bayesian inference (1990), tensor-network representation
- Choice of ensemble
  - extended ensemble method: multicanonical MC (1991, 2001), exchange MC (1996), lifting (2000)…
- Generation of set of candidate configurations
  - non-local (cluster) updates: Swendsen-Wang (1987), Hamiltonian MC (1987), loop (1993), worm (1998)…
- Choice of transition kernel (probabilities)
  - Metropolis, heat bath (Gibbs sampler), over-relaxation (1987), irreversible kernel (2010), event-chain (2013)…
- Algorithm for generating a configuration according to transition probabilities
  - N-fold way (rejection free) (1975), Walker’s method (1977, 2019), order-N algorithm (1995, 2009)…



Many-body wave function and tensor

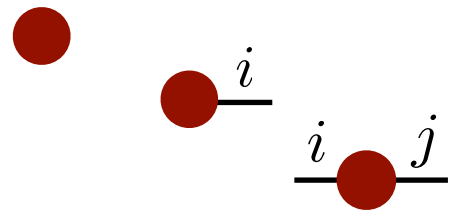
arXiv:2412.02974

- Wave function of N-qubit (spin-1/2) system

$$|\Psi\rangle = \sum_{\sigma_1,\sigma_2,\cdots,\sigma_N} C_{\sigma_1,\sigma_2,\cdots,\sigma_N} |\sigma_1\sigma_2\cdots\sigma_N\rangle$$

  - linear combination of 2^N states → 2^N coefficients($C_{\sigma_1,\sigma_2,\cdots,\sigma_N}$) should be specified → memory cost $\sim 2^N$
  - C can be regarded as N-leg (rank-N) tensor

- Tensor = multi-dim array = generalization of vectors/matrices
  - 0-leg tensor → scalar
  - 1-leg tensor → vector
  - 2-leg tensor → matrix
  - ...
  - N-leg tensor → memory/computational cost $\sim \exp(N)$



Tensor network (tensor diagram)

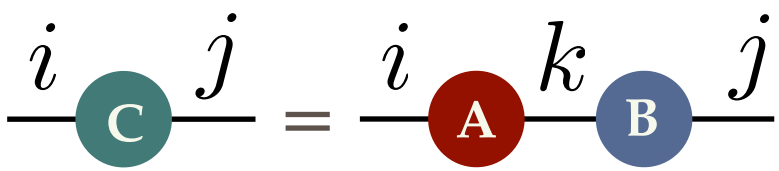
arXiv:2412.02974

- Contraction of tensors
  - taking a summation over shared indices (=connected legs)

- Contraction of two-leg tensors → result is a two-leg tensor
  - matrix-matrix multiplication

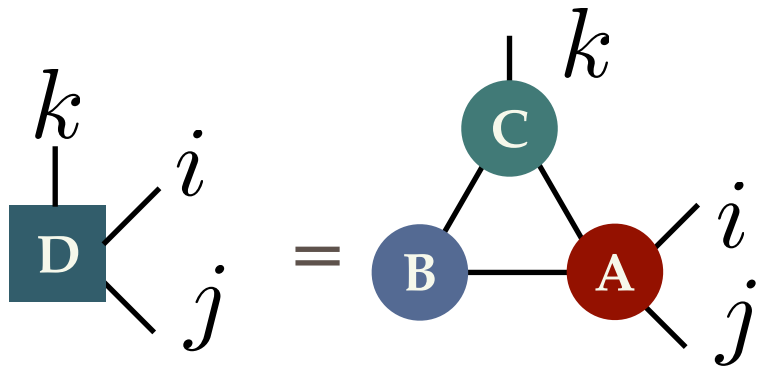
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$

$$C = AB$$



- General tensor contractions can be represented similarly

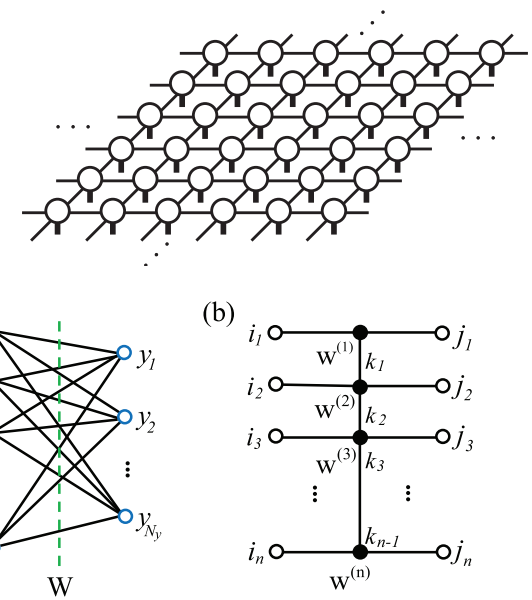
$$D_{i,j,k} = \sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$



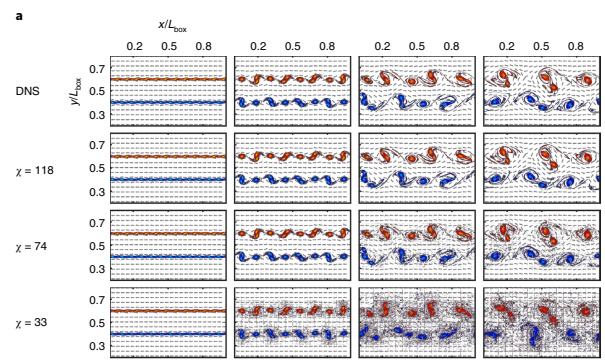
Tensor network representation
テンソルネットワーク表現

arXiv:2412.02974

- Quantum state of quantum many-body systems / 量子多体系の量子状態
  - MPS, Tree TN, MERA, PEPS
  - Sampling Complexity of MPS at finite temperature
- Partition function in statistical physics / 統計力学模型の分配関数
  - TN renormalization / TN繰り込み群
  - Application to lattice field theories / 場の理論への応用
- Machine learning using TN / テンソルネットワークによる機械学習
  - Compression of neural networks/generative models / ニューラルネットワーク・生成モデルの圧縮
- Compression in hierarchical structure / 階層構造の情報圧縮
  - Quantics representation / 同次多項式表現
  - TN simulation of PDE / 偏微分方程式のTNシミュレーション



Gao et al (2020)

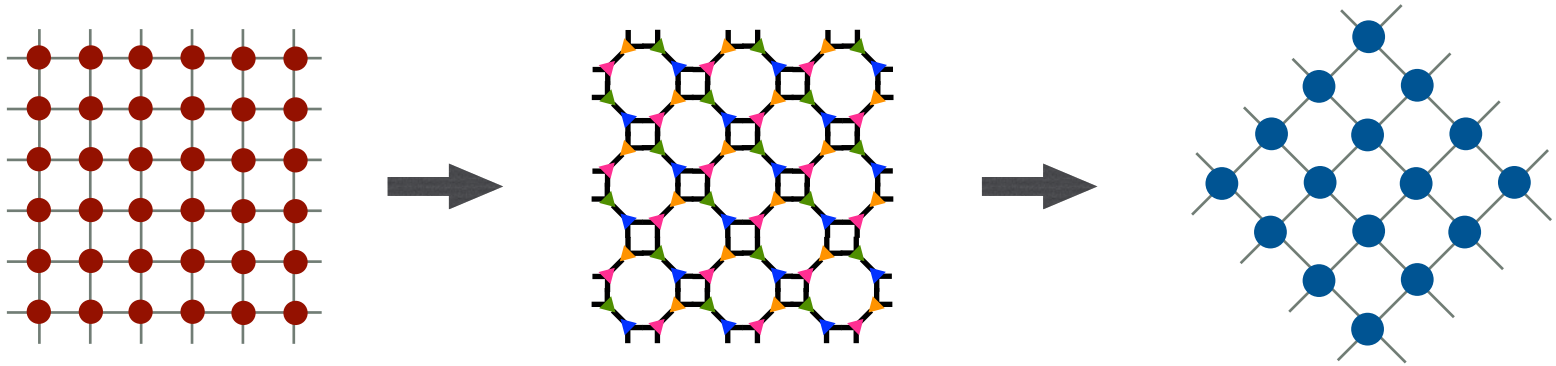


Gourianov et al (2022)

TN methods in statistical physics

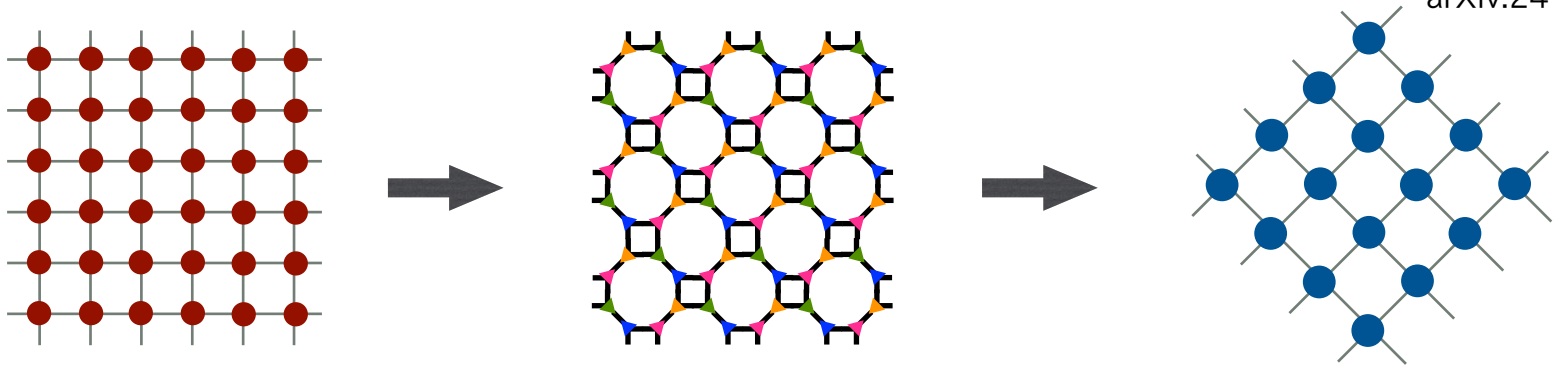
arXiv:2412.02974

- "Renormalization" (or "Lagrangian") approach
  - coarse graining of tensor network representation of the partition function
  - transfer matrix, tensor network renormalization (TRG), higher-order tensor network renormalization (HOTRG), etc
- "Variational" (or "Hamiltonian") approach
  - tensor-network approximation of strongly correlated many-body quantum states
  - DMRG, PEPS, MERA, etc
- "Exact" contraction of tensor network can not be done in two and higher dimensions
  - low-rank approximation based on eigenvalue/singular value decomposition
  - accuracy of approximation is controlled by "bond dimension": D (or χ)

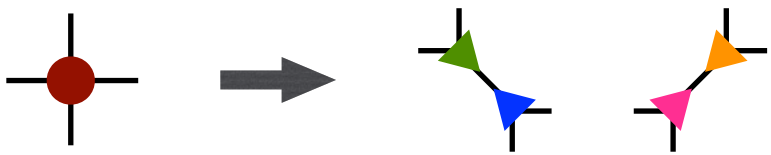


Tensor renormalization group

arXiv:2412.02974



- Low-rank approximation based on SVD



computational cost: $O(D^5)$

memory cost: $O(D^3)$

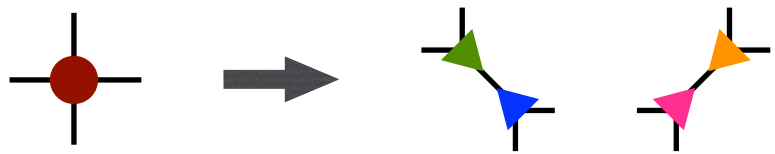
- Improvement of accuracy by considering the “environment” effects or removing local correlations
  - second order renormalization (SRG), mean-field SRG, etc
  - tensor network renormalization (TNR), loop TNR, Gilt, etc
  - computational cost increases significantly

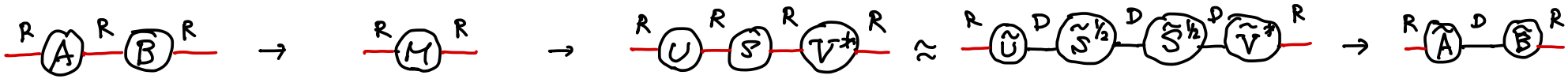
Levin & Nave (2007)

Low-rank approximation in TN methods

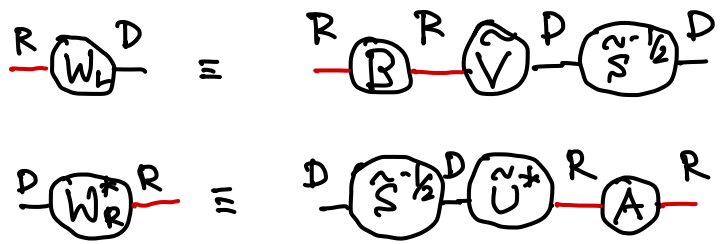
arXiv:2412.02974

- Low-rank approximation based on SVD
  - choose largest d singular values





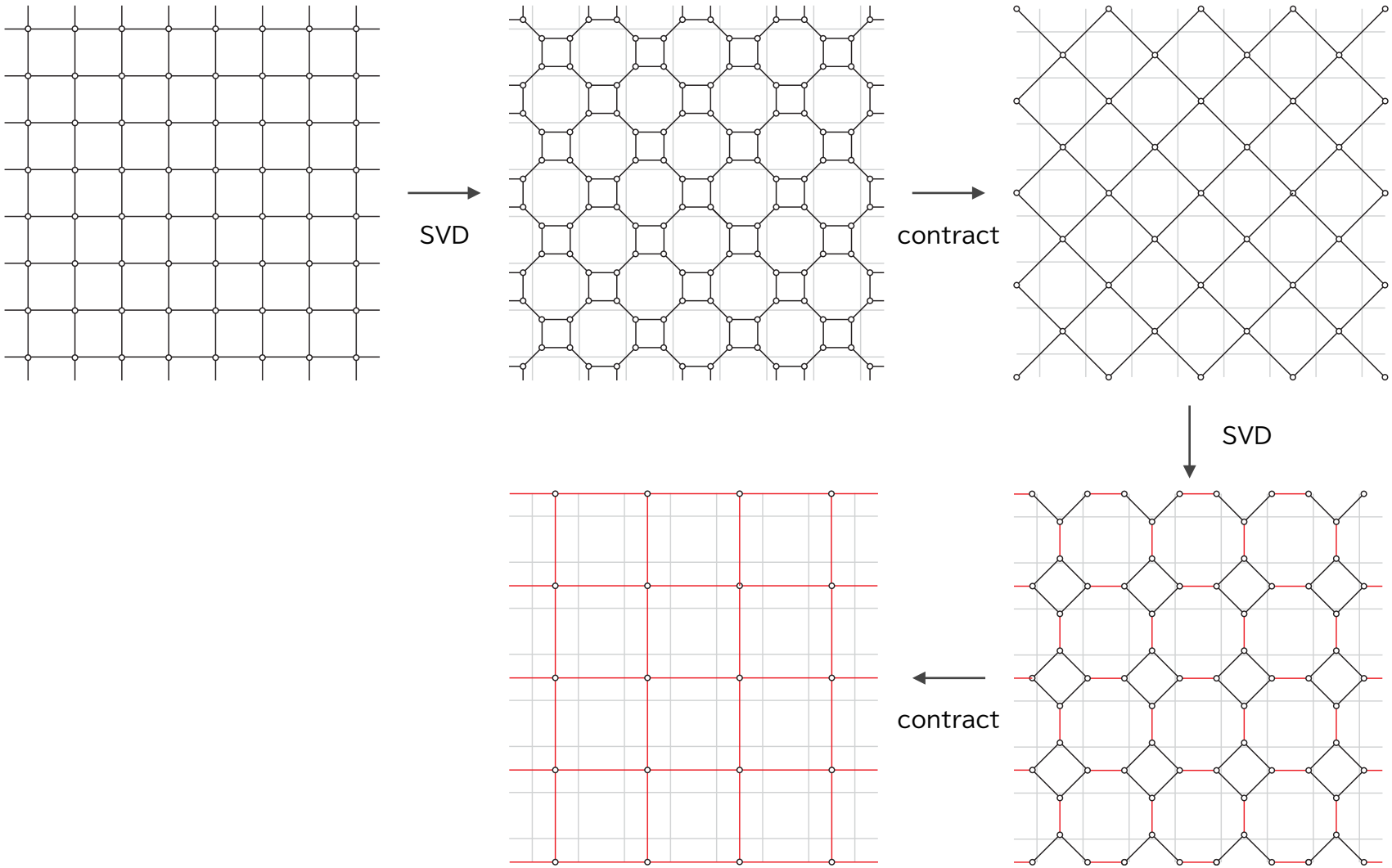
- Projector formulation
  - choose the best projector to d dimensions
  - equivalent to low-rank approximation using SVD



Levin-Nave tensor renormalization group

arXiv:2412.02974

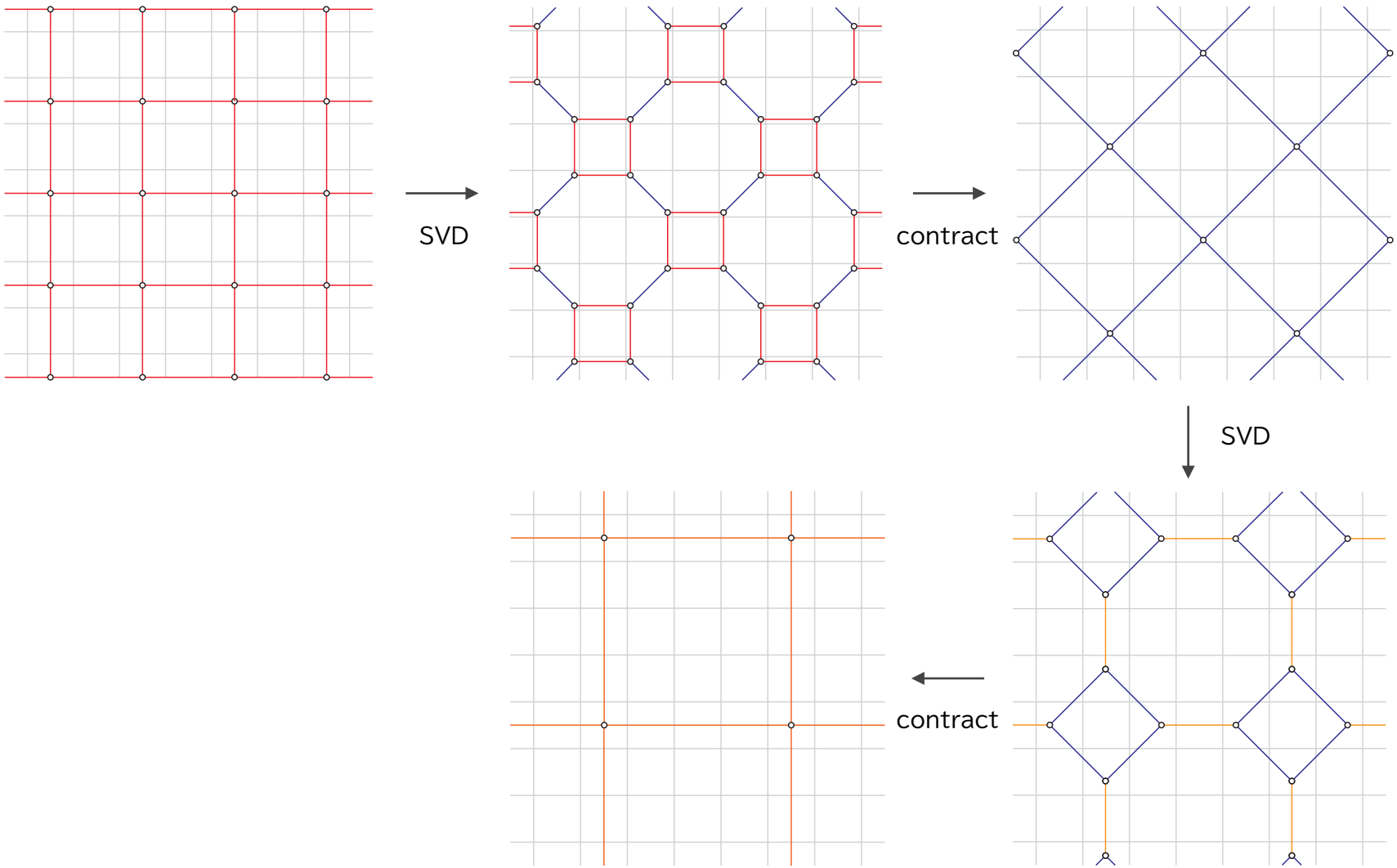
- 8×8 square lattice case ($N = 64$)



Levin-Nave tensor renormalization group

arXiv:2412.02974

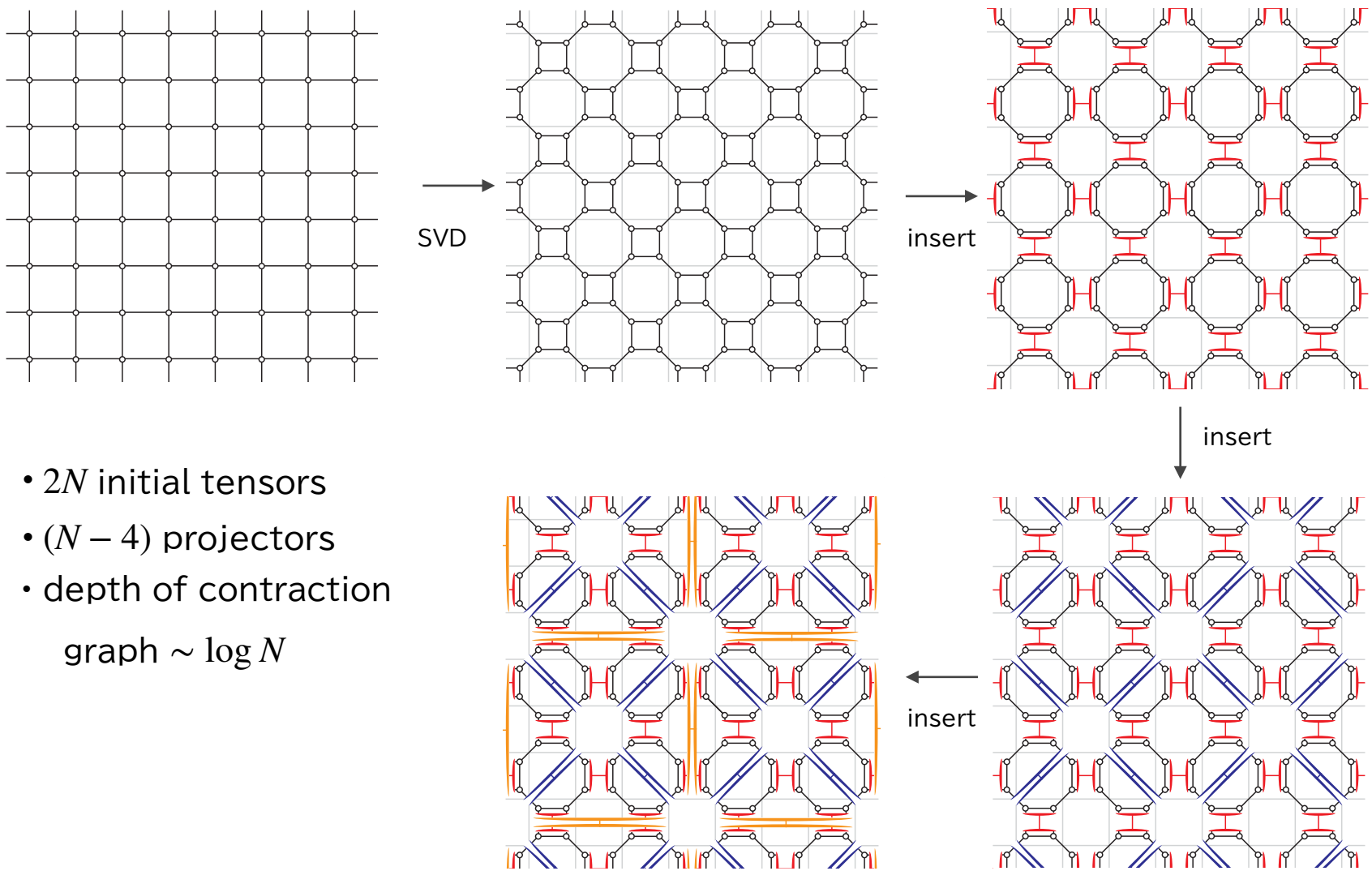
- 8×8 square lattice case ($N = 64$)



Projection formulation of TRG

arXiv:2412.02974

- 8×8 square lattice case ($N = 64$)



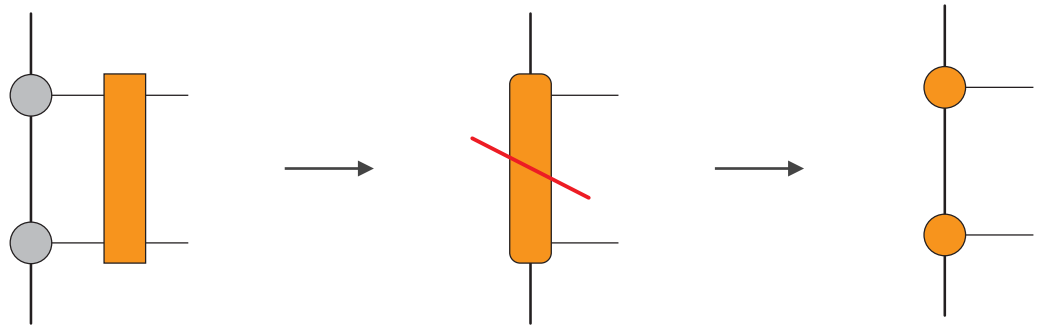
- $2N$ initial tensors
- $(N - 4)$ projectors
- depth of contraction

  graph $\sim \log N$

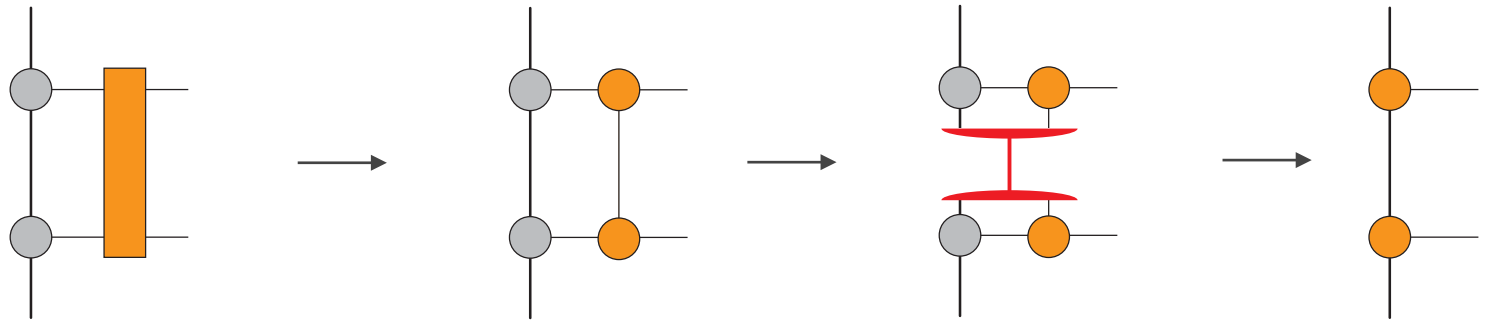
MPS simulation of quantum circuits: TEBD

arXiv:2412.02974

- 2-qubit operation (contract and SVD)



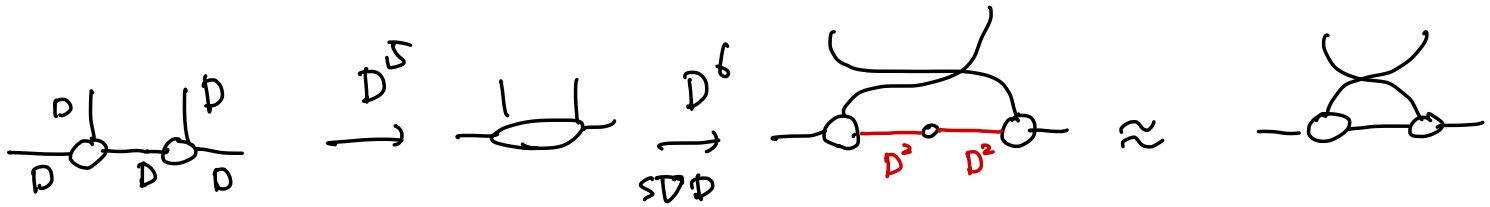
- Projector formulation



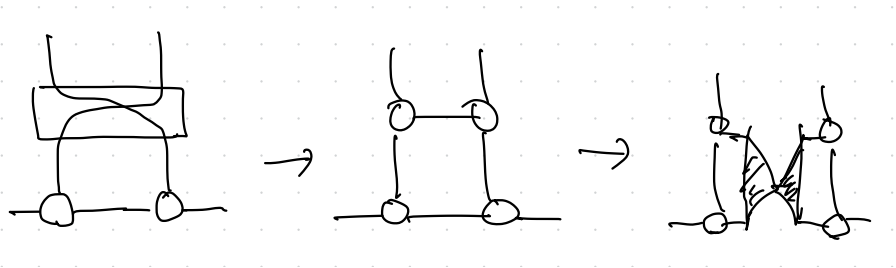
Projector formulation of TN methods

arXiv:2412.02974

- ATRG (Adachi et al 2020) and CATN (Pan et al 2020)
  - leg swap based on SVD



  - leg swap based on projector



- Any tensor network renormalization methods can be reformulated using projectors (?)

Core idea of tensor-network MCMC

arXiv:2412.02974

- Random projectors
  - instead of choosing the “best” projector, all possible projectors are sampled according to some weights, such that on average (Ferris 2015)

$$\langle W_R(\theta) W_L^*(\theta) \rangle_\theta = \frac{1}{n_c} \sum_{\theta=1}^{n_c} W_R(\theta) W_L(\theta) p(\theta) = I_r$$

- Should use Markov chain Monte Carlo to control variance
  - as, in random sampling and importance sampling, statistical error diverges exponentially as system size increases

- Should use different projector configurations in different positions
  - to avoid systematic errors from correlation

Markov chain Monte Carlo approach

arXiv:2412.02974

- Determine projector candidates from SVD during the conventional (deterministic) TRG
  - projectors becomes independent with each other and can be sampled independently

$$p(\theta_1, \theta_2, \cdots, \theta_n) = p(\theta_1)p(\theta_2)\cdots p(\theta_n)$$

  - (exact) tensor network representation of partition function

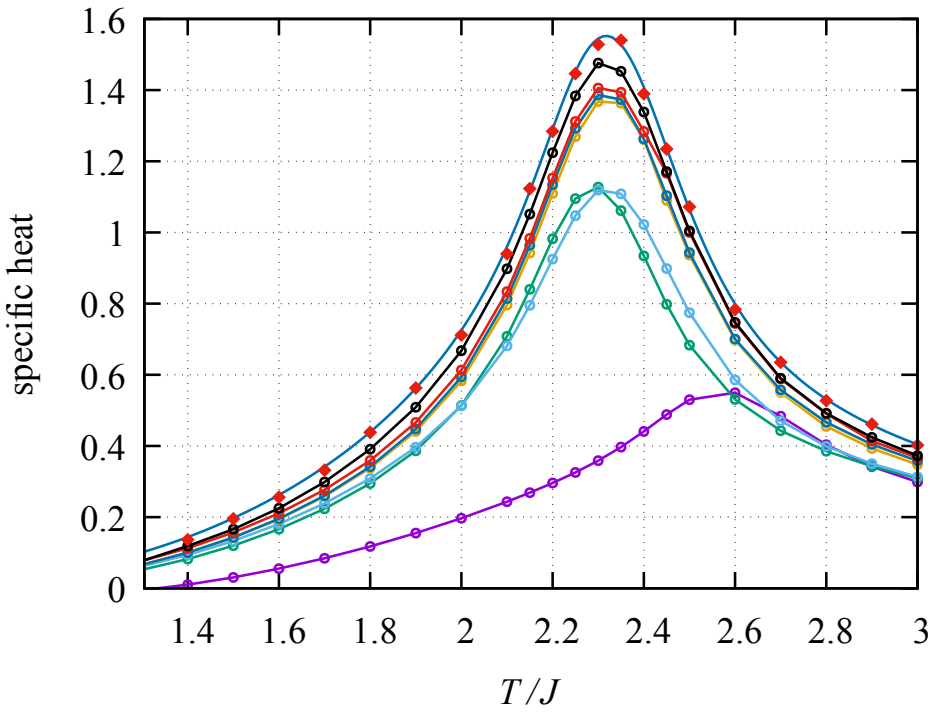
$$Z = \sum_{\{\theta_i\}} g(\theta_1, \theta_2, \cdots, \theta_n)p(\theta_1, \theta_2, \cdots, \theta_n) = \sum_{\{\theta_i\}} g(\theta_1, \theta_2, \cdots \theta_n)p(\theta_1)p(\theta_2)\cdots p(\theta_n)$$

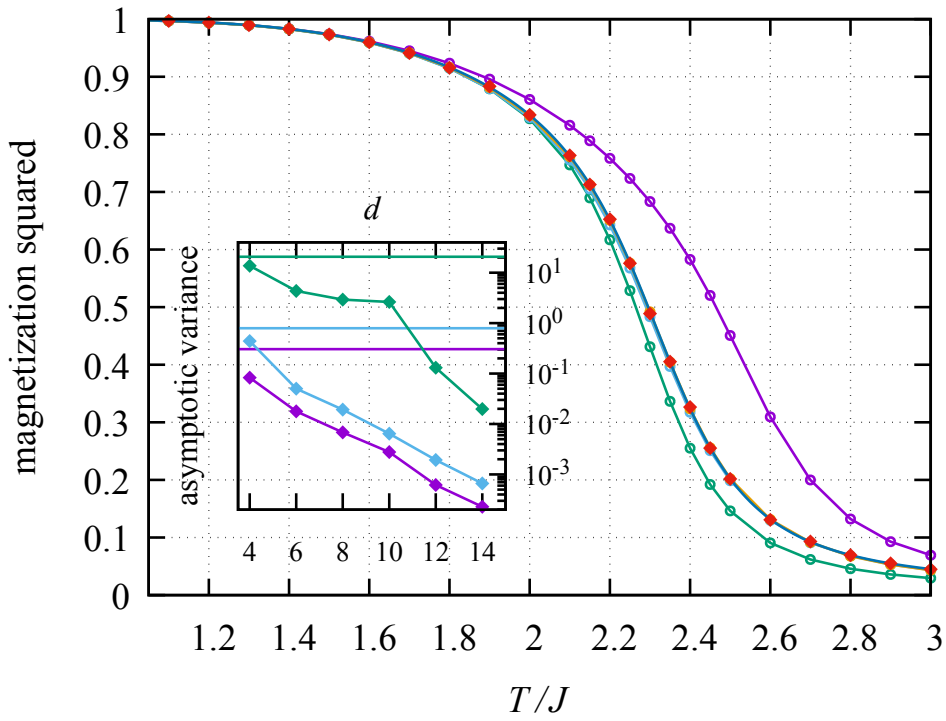
- Sample projectors $\{\theta_i\}$ using Markov-chain Monte Carlo
  - propose new θ_i according to $p(\theta_i)$
  - Metropolis update with $P = \min(1, g(\theta_1, \theta_2, \cdots, \theta_i', \cdots, \theta_n)/g(\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_n))$
  - update of weights is $O(\log N)$ and includes matmul only
  - SVDs are required during the initialization stage only
- Physical quantities
  - can be evaluated by using the impurity tensor technique (without systematic bias)

Comparison with Levin-Nave TRG + impurities

arXiv:2412.02974

- Square-lattice Ising model ($L = 16$)
  - Exact results (transfer matrix): dark blue
  - TensorMC: d=6 (red symbols)
  - Levin-Nave TRG: d=2 (purple), 4 (green), 6 (cyan),..., 16 (black)

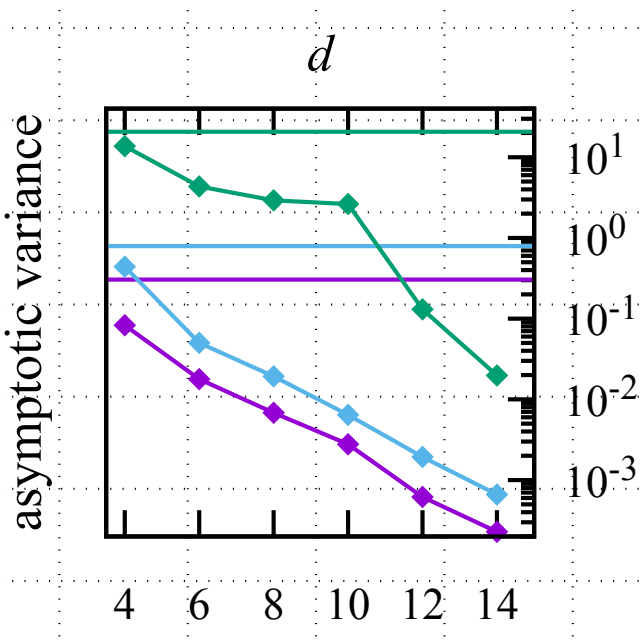




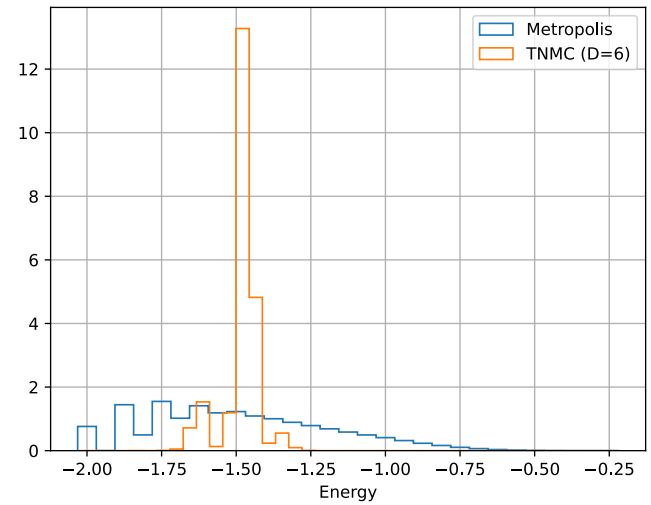
Comparison with Metropolis algorithm

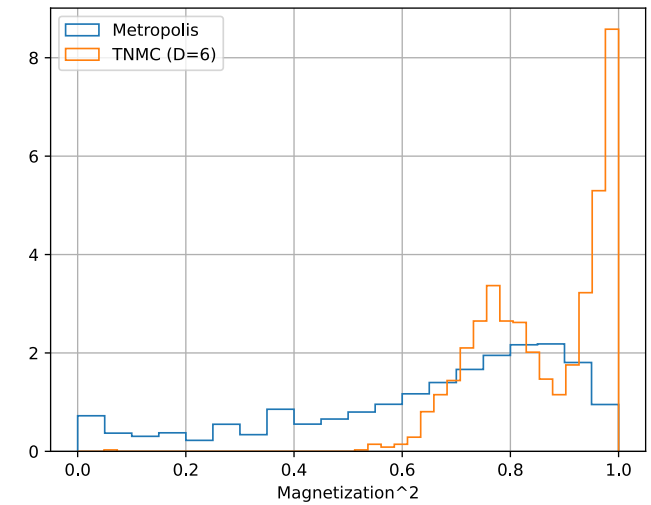
arXiv:2412.02974

- Square-lattice Ising model
  - asymptotic variance is smaller by orders of magnitude
  - asymptotic variance decreases exponentially as d increases



$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\text{int}}}{M}$$





- energy (purple), specific heat (green), magnetization^2 (cyan)

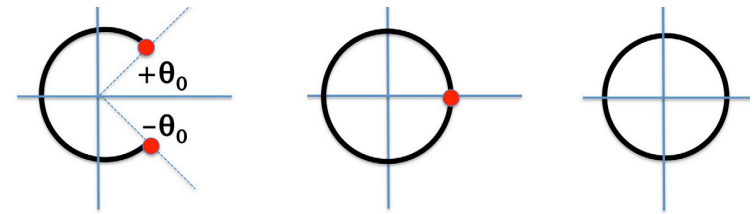
Ising model in imaginary external field

arXiv:2412.02974

- Square lattice Ising model

$$H = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i$$

Yang-Lee zeros on complex plane of fugacity z



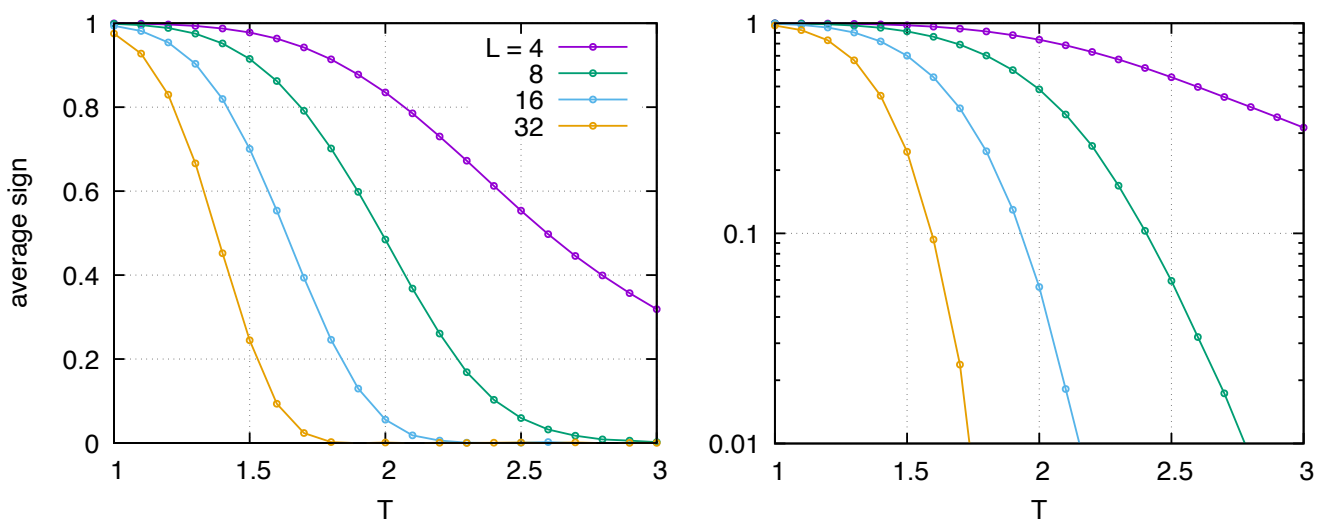
- pure imaginary external field

$$h = i\pi/2\beta \quad \Rightarrow \quad z = e^{-2\beta h} = -1$$

- non-positive Boltzmann weight (m: total magnetization)

$$W = e^{\beta \sum \sigma_i \sigma_j} \times (-1)^{m/2} \qquad (m\text{: total magnetization})$$

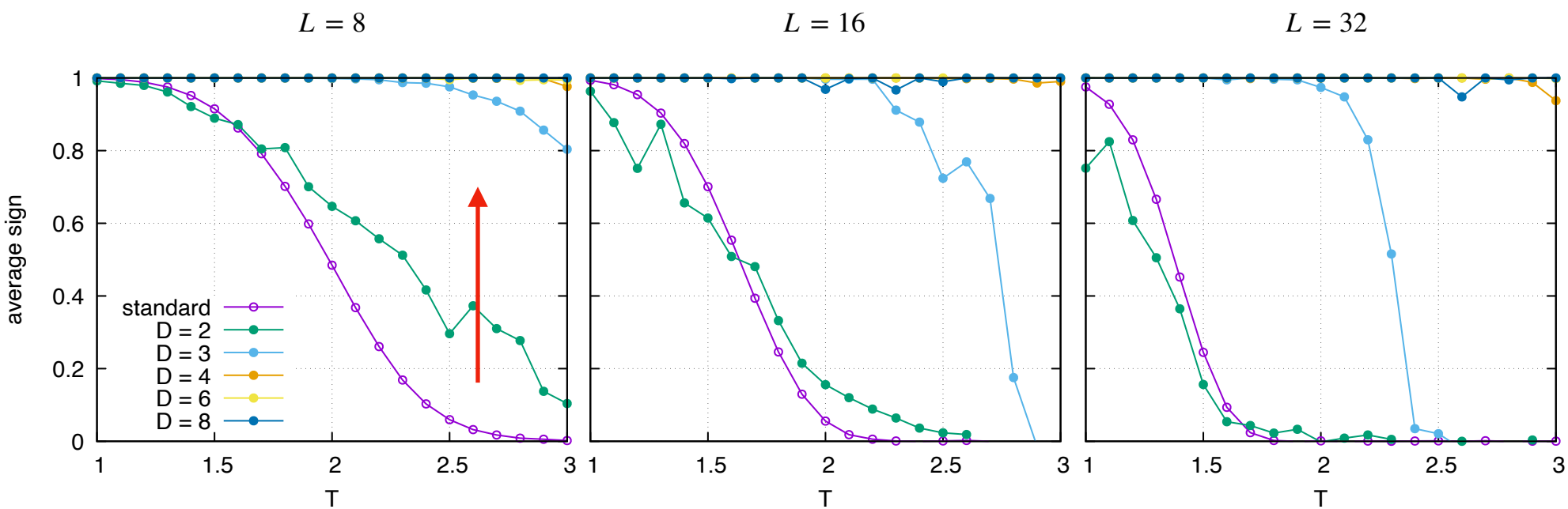
- Standard Markov chain Monte Carlo suffers from severe negative sign problem



Ising model in imaginary external field

arXiv:2412.02974

- Our proposed method also has negative signs for small d
  - $d = 2$ results are almost similar to the standard method
  - NB: negative signs can appear for small d (but not serious) even if the original model is free from negative sign
- However, the average sign is improved drastically as we increase d



Summary and outlook

arXiv:2412.02974

- Markov chain Monte Carlo in tensor network representation
  - reducing statistical error using approximate tensor network contraction
  - removing systematic bias by sampling singular vectors (projectors) using MCMC

  → avoid divergence of statistical error, negative signs, and systematic bias

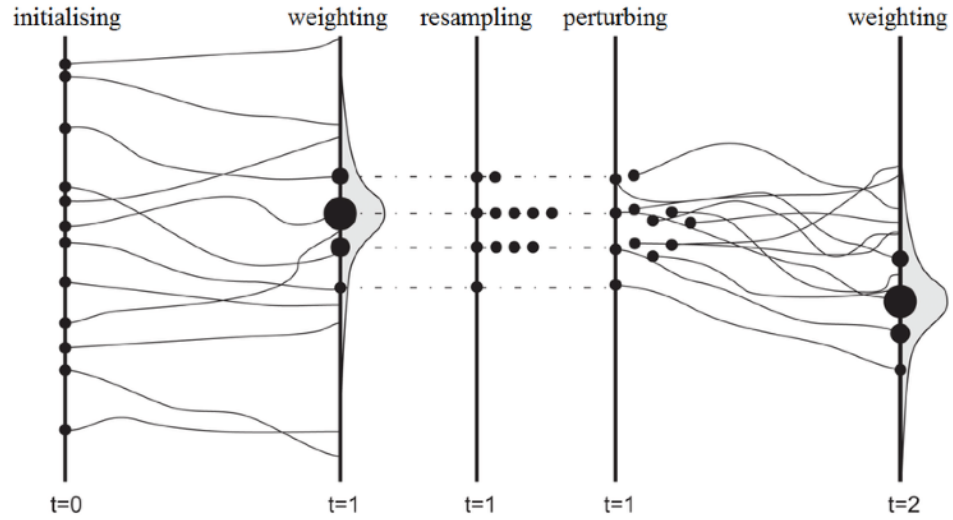
$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\mathrm{int}}}{M}$$

- Computational complexity of one Monte Carlo update
  - $O(d^\alpha N \log N)$
  - matmul only (no SVD) during MCMC sampling → ideal for modern GPGPU or HPC
- Combination with various advanced sampling techniques (to reduce τ_{int})
- Applications:
  - quantum spin models (via Suzuki-Trotter decomposition)
  - higher dimensions: HOTRG (2012), ATRG (2020)
  - fermions, quantum circuits, etc
  - variational MC based on PEPS
  - others?

Combination with Sequential MC

arXiv:2412.02974

- Sequential MC aka
  - green's function Monte Carlo
  - transfer matrix Monte Carlo
  - population Monte Carlo
  - Monte Carlo filter
  - particle filter
  - bootstrap filter
  - SISR (sequential importance sampling with resampling)



https://www.researchgate.net/figure/Sequential-Monte-Carlo-scheme_fig2_322302619

- Central idea of SMC
  - approximate probability distribution by (weighted) ensemble of particles
  - by using Markov chain
  - control variance by resampling

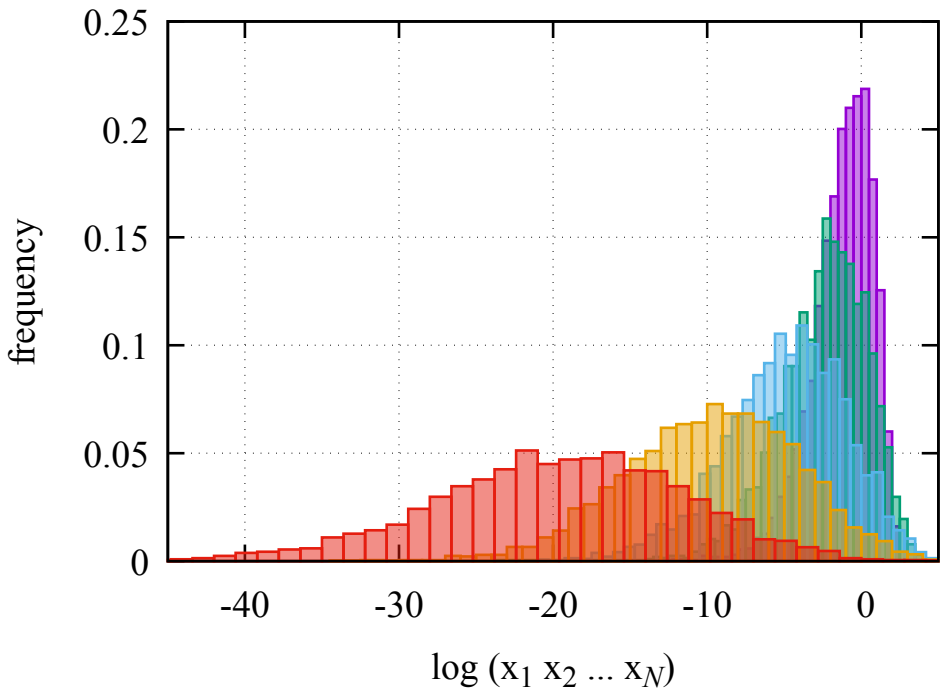
A simple example

arXiv:2412.02974

- estimate mean of product of N random numbers:

$$\mathrm{E}[X_1 X_2 \cdots X_N] = \Gamma$$

- x_k is sampled independently randomly from uniform distribution between 0 and 2
- expectation value: $\Gamma = 1$
- typical value of product $\sim \exp[(\log 2 - 1)N]$
- variance increases rapidly for large N (4, 8, 16, 32, 64)



$$\mathrm{Var}[X_1 X_2 \cdots X_N] \sim (4/3)^N$$

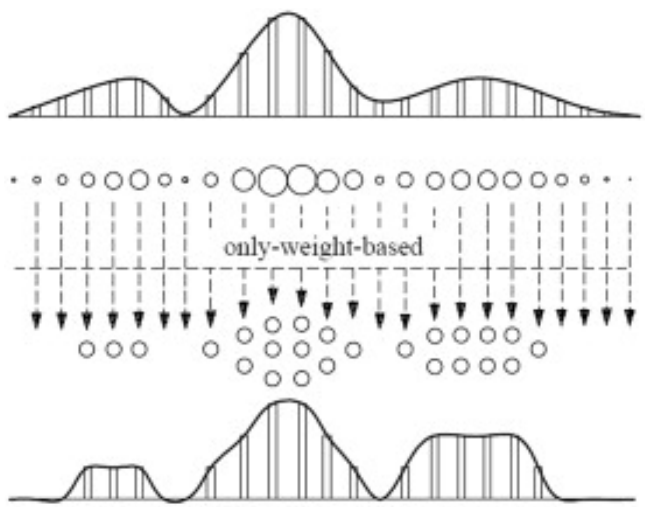
Resampling

arXiv:2412.02974

- Simple sequential importance sampling becomes unstable for large steps
  - weight of each walker is updated randomly by weight factors: $W = w_1 w_2 w_3 \cdots$
  - random walk diffusion in logarithmic scale
  - weight degeneracy: weight variance (discrepancy between weights) grows exponentially and only a few walkers dominate
- Resampling is necessary to stabilize the algorithm
  - resampling:

$$P_i \simeq \sum_k W_k \delta_{i,i_k} \Rightarrow \sum_k \delta_{i,\tilde{i}_k}$$

  - after resampling, all walkers share the same weight: $\sum_k W_k / N_w$



Li-Stattar-Sun (2012)

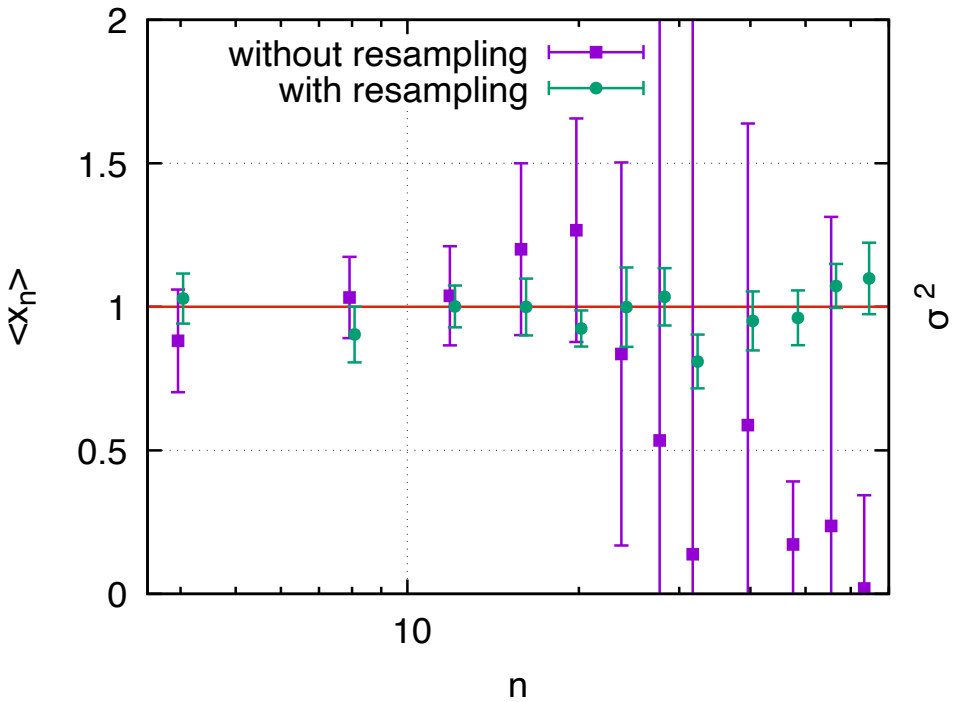
Effect of resampling

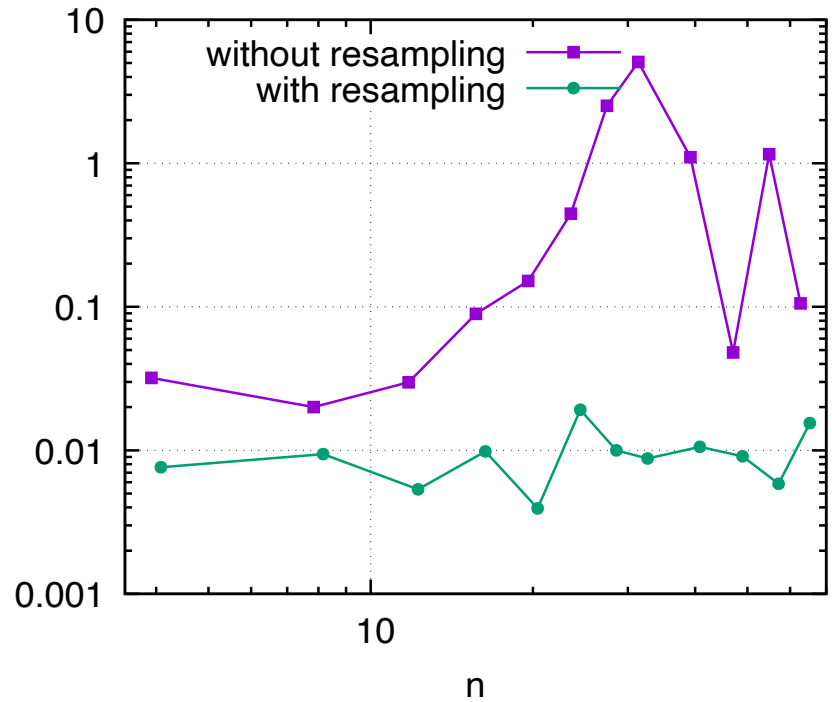
arXiv:2412.02974

$$x_0 = 1$$

$$x_k = \xi_k x_{k-1} \quad (k = 1,2,\cdots,n)$$

- ξ_k is sampled independently randomly from uniform distribution between 0 and 2





Summary and outlook

arXiv:2412.02974

- 5 steps to MCMC-ize your TRG algorithms
  1. Select the TRG method most appropriate for your target.
  2. Rewrite it into a projector formalism (replace SVD by projector insertion)
     - This step is a bit non-trivial
  3. Perform (deterministic and optimal) TRG method to collect a complete set of rank-1 projectors and their weights (singular values)
     - We have already developed a library for this purpose (but in Python)
  4. If your TRG method assumes translational symmetry (an O(log N) method), convert it to an O(N) procedure that does not assume translational symmetry
     - We have a prototype library that performs this conversion automatically (but in Python)
  5. Perform MCMC sampling
     - We have a generic prototype code for MCMC sampling (but in Python)

Summary and outlook

arXiv:2412.02974

- Developing a standard tensor network library
  - low-level, high-performance, portable, stable, and clean
  - gather good experiences from many existing libraries, and avoid bad practices
  - implemented in Rust, then wrappers for C/C++, Fortran, Python, Julia, etc
  - explicit memory management
  - thread-safe, MPI parallelization, GPU support
  - tensor contraction without transpose
  - support diagonal tensors, Grassmann tensors
  - + mid-level interfaces
    - tensor SVD, tensor QR, tensor functions, etc
  - + TensorMC support
    - projector generation, conversion into O(N) procedure, projector sampling, contraction graph and cache mechanism

- The name is still a secret...